IN THE CLAIMS

1. (Currently Amended) An Application Programming Interface (API), comprising:

a non-scriptable plug-in API;

a scriptable plug-in API; and

a plurality of bridges operatively configured to connect~~ing~~ said scriptable and said non-scriptable plug-in APIs such that a scriptable plug-in program is able to access the non-scriptable plug-in API in response to implementing the scriptable plug-in API.

2. (Currently Amended) The API of claim 1, wherein: ~~further comprising:~~

~~a first interface in said~~ the scriptable plug-in API has a plurality of first interfaces; ~~and~~

~~a second interface in said~~ the non-scriptable plug-in API has a plurality of second interfaces; ~~wherein one~~ and

each of said bridges connects a respective one of said first interfaces ~~and~~ to a respective one of said second interfaces.

3. (Currently Amended) An API comprising:

a cross platform language API;

a scriptable language API; ~~and~~

~~an~~ a first interface ~~between~~ operatively configured to connect said cross platform language API and said scriptable language API;

a non-scriptable plug-in API; and

a second interface operatively configured to connect said non-scriptable plug-in API and said cross platform language API such that said scriptable language API is able to access said non-scriptable plug-in API.

4. (Cancelled Without Prejudice)

5. (Currently Amended) The API of claim [4]3 further comprising:

a non-scriptable plug-in object implemented in said non-scriptable plug-in API;

a cross platform language object implemented in said cross platform language API to correspond to said non-scriptable plug-in object; and

a scripting language object implemented in said scripting language API;

~~a non-scriptable plug-in object in said non-scriptable plug-in API;~~

wherein said second interface operatively connects said non-scriptable plug-in object and said cross platform language object, and

~~an~~ said first interface ~~between~~ operatively connects said cross platform language object and said scripting language object, such that said cross platform language object operates as a proxy for said non-scriptable plug-in object~~; and~~

~~an interface between said non-scriptable plug-in object and said cross platform language object~~.

6. (Currently Amended) The API of claim [4]3, wherein said first interface is an XPIDL interface.

7. (Currently Amended) The API of claim 3, wherein said second interface is an XPConnect interface.

8. (Currently Amended) The API of claim 7, wherein said XPConnect interface uses a typelib file~~s~~ associated with the cross platform language object.

9. (Currently Amended) The API of claim 3 wherein said scripting language object is one of a Javascript object, a Perl object, ~~or~~ and a Python object.

10. (Currently Amended) The API of claim 3, wherein said cross platform language is XPCOM.

11. (Currently Amended) A scriptable plug-in API comprises:

a scriptable plug-in; and

a proxy support interface, wherein said scriptable plug-in ~~can~~ is able to perform inter-thread calls through said proxy support interface.

12. (Currently Amended) The scriptable plug-in API of claim 11, wherein said proxy support interface is a nsISupports Proxy.

13. (Currently Amended) A method for implementing an API, comprising:

obtaining a non-scriptable plug-in API;

obtaining a scriptable plug-in API; and

implementing a plurality of bridges to operatively connect said scriptable and said non-scriptable plug-in APIs such that a scriptable plug-in program is able to access the non-scriptable plug-in API in response to implementing the scriptable plug-in API.

14. (Currently Amended) The method of claim 13, wherein: ~~further comprising:~~

~~defining a first interface in~~ said scriptable plug-in API has a plurality of first interfaces; ~~and~~

~~defining a second interface in~~ said non-scriptable plug-in API has a plurality of second interfaces; and

~~using one~~ each of said bridges ~~to~~ connects a respective one of said first interfaces ~~and~~ to a respective one of said second interfaces.

15. (Currently Amended) A method for implementing an API comprising:

obtaining a cross platform language API;

obtaining a scriptable language API; and

implementing ~~an~~ a first interface ~~between~~ to operatively connect said cross platform language API and said scriptable language API;

obtaining a non-scriptable plug-in API; and

implementing a second interface to operatively connect said non-scriptable plug-in API and said cross platform language API such that said scriptable language API is able to access said non-scriptable plug-in API.

16. (Cancelled without prejudice)

17. (Currently Amended) The method of claim ~~16~~15 further comprising:

defining a non-scriptable plug-in object in said non-scriptable plug-in API;

defining a cross platform language object in said cross platform language API to correspond to said non-scriptable plug-in object; and

defining a scripting language object in said scripting language API;

~~defining a non-scriptable plug-in object in said non-scriptable plug-in API;~~

~~implementing an interface between said cross platform language object and said cross platform language object.~~

wherein said second interface operatively connects said non-scriptable plug-in object and said cross platform language object, and

said first interface operatively connects said cross platform language object and said scripting language object, such that said cross platform language object operates as a proxy for said non-scriptable plug-in object.

18. (Currently Amended) The method of claim ~~16~~15, wherein said first interface is an XPIDL interface.

19. (Currently Amended) The method of claim 15, wherein said second interface is an XPConnect interface.

20. (Currently Amended) The method of claim 19, wherein said XPConnect interface uses a typelib file~~s~~ associated with the cross platform language object.

21. (Currently Amended) The method of claim 15, wherein said scripting language object is one of a Javascript object, a Perl object, ~~or~~ and a Python object.

22. (Currently Amended) The method of claim 15, wherein said cross platform language is XPCOM.

23. (Currently Amended) A method of implementing a scriptable plug-in API, the method comprising;

implementing a scriptable plug-in; and

~~using~~ generating a proxy support interface wherein said scriptable plug-in ~~can~~ is able to perform a inter-thread call~~s~~ through said proxy support interface.

24. (Currently Amended) The method of claim 23, wherein said proxy support interface is a nsISupports Proxy.

25. (Currently Amended) A computer ~~program product~~ readable medium containing instructions causing a program in a computer system to perform a method, the method comprising:

~~a computer usable medium having computer readable program code embodied therein configured to implement an API, said computer program product comprising:~~

~~computer readable code configured to cause a computer to~~ obtaining a non-scriptable plug-in API;

~~computer readable code configured to cause a computer to~~ obtaining a scriptable plug-in API; and

~~computer readable code configured to cause a computer to~~ implementing a plurality of bridges to operatively connect said scriptable and said non-scriptable plug-in APIs such that a scriptable plug-in program is able to access the non-scriptable plug-in API in response to implementing the scriptable plug-in API.

26. (Currently Amended) The computer ~~program product~~ readable medium of claim 25, wherein: ~~further comprising:~~

~~computer readable code configured to cause a computer to define a first interface in~~ said scriptable plug-in API has a plurality of first interfaces;

~~computer readable code configured to cause a computer to define a second interface in~~ said non-scriptable plug-in API has a plurality of second interfaces; and

~~computer readable code configured to cause a computer to use one~~ each of said bridges to connect said first interface and said second interface.

27. (Currently Amended) A computer ~~program product~~ readable medium containing instructions causing a program in a computer system to perform a method, the method comprising:

~~a computer usable medium having computer readable program code embodied therein configured to implement an API, said computer program product comprising:~~

~~a computer readable code configured to cause a computer to~~ obtaining a cross platform language API;

~~computer readable code configured to cause a computer to~~ obtaining a scriptable language API; and

~~computer readable code configured to cause a computer to~~ implementing ~~an~~ a first interface ~~between~~ to operatively connect said cross platform language API and said scriptable language API;

obtaining a non-scriptable plug-in API; and

implementing a second interface to operatively connect said non-scriptable plug-in API and said cross platform language API such that said scriptable language API is able to access said non-scriptable plug-in API.

28. (Cancelled without prejudice)

29. The computer ~~program product~~ readable medium of claim ~~28~~27, the method further comprising:

defining a non-scriptable plug-in object in said non-scriptable plug-in API;

~~computer readable code configured to cause a computer to define~~ defining a cross platform language object in said cross platform language API to correspond to said non-scriptable plug-in object; and

~~computer readable code configured to cause a computer to define~~ defining a scripting language object in said scripting language API;

~~computer readable code configured to cause a computer to define a non-scriptable plug-in object in said non-scriptable plug-in API;~~

~~computer readable code configured to cause a computer to implement an~~, wherein said first interface ~~between~~ operatively connects said cross platform language object and said scripting language object; and

~~computer readable code configured to cause a computer to implement an~~ said second interface ~~between~~ operatively connects said non-scriptable plug-in object and said cross platform language object, such that said cross platform language object operates as a proxy for said non-scriptable plug-in object.

30. (Currently Amended) The computer ~~program product~~ readable medium of claim ~~28~~27, wherein said first interface is an XPIDL interface.

31. (Currently Amended) The computer ~~program product~~ readable medium of claim 27, wherein said second interface is an XPConnect interface.

32. (Currently Amended) The computer ~~program product~~ readable medium of claim 31, wherein XPConnect interface uses a typelib file~~s~~ associated with the cross platform language object.

33. (Currently Amended) The computer ~~program product~~ readable medium of claim 27, wherein said scripting language object is one of a Javascript object, a Perl object, ~~or~~ and a Python object.

34. (Currently Amended) The computer ~~program product~~ readable medium of claim 27, wherein said cross platform language is XPCOM.

35. (Currently Amended) A computer ~~program product~~ readable medium containing instructions causing a program in a computer system to perform a method, the method comprising:

~~a computer usable medium having computer readable program code embodied therein configured to implement a scriptable plug-in API, said computer program product comprising:~~

~~a computer readable code configured to cause a computer to~~ obtaining a scriptable plug-in; and

~~a computer readable code configured to cause a computer to use~~ generating a proxy support interface ~~wherein~~ such that said scriptable plug-in ~~can~~ is able to perform an inter-thread call~~s~~ through said proxy support interface.

36. (Currently Amended) The computer ~~program product~~ readable medium of claim 35, wherein said proxy support interface is a nsISupports Proxy.